AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph at page 14, lines 8-18, with the following paragraph:

The presently preferred algorithm proceeds through the following steps:

1. Start from the deepest active list in the lexical tree.
2. Let B be the smallest ranked node in the active list of the children column.
3. Traverse the active list in increasing order.
4. For each child c of the current node k,
5. If B < c, then increment B until that condition is false.
6. If B = c, then apply the dynamic programming equation.
7. If B > c, then simply link c before n.
8. Decrement the depth and process the parent column.

* n: the lowest rank active node in a given level. In Figure 4b, it would be respectively the node pointed to by arrows numbered 0, 2, 4, 6, and 7.

* c: denotes a child on current node k. For instance, in Figure 4b, node [H]'s only child would be [AA]. This node's children would be [R] and [L]. The "c" would assume the values "R" and "L" (in that order).

* k: the current node in the active list. In the Figure 4b, it assumes all nodes pointed to by arrows 0, 1, 2, etc, successively.

Please replace the paragraph at page 17, lines 1-11, with the following paragraph:

Figure 4a shows an example where words, instead of letters, are represented at each node. In the preceding examples, an individual word recognizer was illustrated. Each node of the tree represented a letter or sound unit that comprises a word in the dictionary. However, it will be recalled that the techniques of the invention can be used in both individual word and continuous speech recognizers. Thus Figure 4a shows how the tree structure might look in a continuous speech recognizer, in which individual sound units ~~words~~ are represented at each node and the output would be words~~sentences or phrases~~. By examining the tree **70** in Figure 4a one can see, for example, how the word "card"~~phrase "the quick brown fox"~~ would be constructed by appropriate traversal of the tree.

Please replace the paragraph at page 17, line 12 – page 18, line 3 with the following paragraph.

Figure 4a shows how the active node envelope will appear to propagate over time. Timeline **72** shows how the next active node envelope for the exemplary tree might appear at a first time a, and at a later time b. Time a corresponds to the point within the utterance ~~"the quick brown fox"~~ "card" immediately after the sound unit "k" ~~word "the"~~ has been analyzed by the speech analysis step **10** (Fig. 1). Time b corresponds to the point at which the sound "r" ~~word "brown"~~ has been processed. At

time a the active envelope is illustrated at **74**, correspond to those that are most likely to match the utterance that has been partially analyzed at this point. At later time b the active envelope has propagated outwardly, as illustrated at **76**. The active node envelopes at **74** and at **76** represent the active nodes at two different points in time (time a and time b). The algorithm operates upon these active nodes, using the currently active nodes to define the entry point into the lexical tree for the next successive iteration.